A GENERAL GOAL POINT MODEL FOR ANTICIPATION OF VEHICLE MOTIONS FOR AUTONOMOUS DRIVING

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master

by

Yichen Zhou

May 2019

ABSTRACT

In the field of autonomous driving, anticipation of the dynamic environment is of great importance for the ego vehicle to make decisions and plan future paths in order to ensure safety and efficiency. This thesis presents a general goal point model for making predictions of vehicle motions around a moving ego vehicle. One or multiple goal points are selected based on a road graph and other environmental information. Vehicle predictions are then initialized from a probabilistic tracker and propagated via a motion model toward the goal point. This anticipation model is validated on a real-time dataset and evaluated against an open-loop, purely kinematic baseline model, demonstrating its predictive performance over a 1.5-second window in various scenarios.

BIOGRAPHICAL SKETCH

Yichen Zhou acquired his B.E. degree in Mechanical Engineering at Tsinghua University in Beijing, China in 2015, where he focused on precision control design for frequency-varying contour tracking for his thesis. In the summer of 2014, Yichen was a visiting student researcher at University of California, Berkeley and worked on the project of iterative learning control for wafer scanning systems.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Professor Mark Campbell for his support and patience during the time of my graduate study and related research. He has guided me through this project and in writing of this thesis.

In addition to my advisor, I would like to render my appreciation to my special committee member, Professor Ross Knepper, and my former special committee member, Professor Kilian Weinberger for their support, encouragement and cooperation.

I also want to thank my fellow labmates for discussing academic problems with me, helping me collect data and debug programs, and even offering me life advice.

At last, I would like to thank my family for supporting me throughout this whole experience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

time variables

t_i current time stamp ($i = 0$ stands for current one)

k_i index for time step ($i = 0$ stands for current one)

t_p prediction time span

t_s sampling time for prediction

state variables

x X-axis position of target vehicle in ego vehicle coordinate, represented by midpoint of back axle

y Y-axis position of target vehicle in ego vehicle coordinate, represented by midpoint of back axle

s speed of target vehicle in an inertial coordinate attached to the ground

ϕ heading of target vehicle in ego vehicle coordinate

l length of the rectangle approximating the shape of target vehicle

w width of the rectangle approximating the shape of target vehicle

a acceleration of target vehicle in an inertial coordinate attached to the ground

Ω angular velocity of target vehicle in an inertial coordinate attached to the ground

state vectors

$\mathbf{q}_k$ basic state vector $[x\ y\ s\ \phi]^\mathsf{T}$ at time step k

$\mathbf{q}_k^a$ augmented state vector $[x\ y\ s\ \phi\ l\ w\ a\ \Omega]^\mathsf{T}$ at time step k

$\hat{\mathbf{q}}_k$ estimation of $\mathbf{q}_k$

$\hat{\mathbf{q}}_k^a$ estimation of $\mathbf{q}_k^a$

$\tilde{\mathbf{q}}_k$ prediction of $\mathbf{q}_k$

odometry variables

v_x component of ego vehicle's velocity in an inertial coordinate attached to the ground along x-axis of ego vehicle coordinate

v_y component of ego vehicle's velocity in an inertial coordinate attached to the ground along y-axis of ego vehicle coordinate

ω_z rotational speed of ego vehicle in an inertial coordinate attached to the ground

measurements

$\mathcal{L}_k$ clustered LiDAR points at time step k stored in ego coordinate

b_{min} minimum counterclockwise bearing of target vehicle's occlusion in ego vehicle coordinate

b_{max} maximum counterclockwise bearing of target vehicle's occlusion in ego vehicle coordinate

r_{min} minimum range of target vehicle's occlusion in ego vehicle coordinate

$b_{r_{min}}$ bearing corresponding to the location of minimum range r_{min} in ego vehicle coordinate

$\mathbf{z}_k$ meta-measurement vector $[b_{min}\ b_{max}\ r_{min}\ b_{r_{min}}]^\mathsf{T}$ at time step k

functions and noises for estimation

$\mathbf{f}^a(\cdot,\cdot)$ dynamic function for estimation of $\mathbf{q}^a$

$\mathbf{h}(\cdot,\cdot)$ observation function for estimation of $\mathbf{q}^a$

$\boldsymbol{\mu}^a$ process noise vector, $\boldsymbol{\mu}^a = [\mu_x^a\ \mu_y^a\ \mu_s^a\ \mu_\phi^a\ \mu_l^a\ \mu_w^a\ \mu_a^a\ \mu_\Omega^a]^\mathsf{T}$

$\boldsymbol{\nu}$ measurement noise vector, $\boldsymbol{\nu} = [\nu_{b_{min}}\ \nu_{b_{max}}\ \nu_{r_{min}}\ \nu_{b_{r_{min}}}]^\mathsf{T}$

functions and variables for prediction

$\mathbf{f}^p(\cdot,\cdot)$ dynamic function for prediction of $\mathbf{q}$

$\mathbf{u}_c$ control input vector for assumed acceleration and angular velocity

a_c assumed acceleration for prediction of $\mathbf{q}$

Ω_c assumed angular velocity for prediction of $\mathbf{q}$

variables and notations for goal point anticipation model

S starting point of prediction

G goal point for prediction

$\vec{v}_0$ velocity vector of target at S

d_0 lane segment offset of S

$\vec{v}_f$ velocity vector assigned to G

d_f lane segment offset of G

l desired length for target to travel during prediction

C center of arc tangent to two disconnected lane segments

P projection of starting point onto the center reference line or arc of a partition

w_i weighted distance metric to determine possible initial road partitions for vehicles at intersections

p_j estimated probability for each of multiple possible paths in one prediction

CHAPTER 1

INTRODUCTION

1.1 Motivation

Autonomous driving has attracted significant attention over the past ten years, due to its potential to make vehicle traveling safer and more efficient. Decision making and path planning are two important stages that determine the motion behavior of autonomous vehicles. These modules are counted on for vehicles to reach destinations within desired time frame, provide passengers with comfortableness, observe traffic rules, and avoid accidents.

With perception and object tracking techniques, modules of decision making and path planning can obtain an understanding of the environment and function accordingly. However, if no predictions of the dynamic scene are conducted, such modules become reactionary and must reply on rapid replanning to respond to any critical external changes. This approach may not be fast enough and thus fail at times. One related example is the collision between MIT and Cornell in DARPA Urban Challenge in 2007 [1], in which neither cars anticipated the actions of the other one. Team Cornell later revisited the scenario using logged data from then and showed that this MIT-Cornell collision could have been predicted and therefore prevented by the proposed anticipation algorithm from [2]. Anticipation of possible motions of surrounding objects can strongly benefit decision making and path planning to make driving more reliable, robust and capable of tackling all types of scenarios.

Among all types of dynamic objects in the environment surrounding an au-

tonomous car, vehicle is one of the most important categories due to driving dependencies. This thesis focuses on anticipation of vehicle motions for autonomous driving.

1.2 Related Work

Much research has been conducted in the area of anticipation of driver behavior and vehicle trajectory. Some adopt statistic time series modeling techniques without dynamics modeling, such as Hidden Markov Model (HMM), Dynamic Bayesian Network (DBN) and Gaussian Mixture Model (GMM), to model driving behavior [3] [4] [5], to infer turning intention [6], and to predict vehicle trajectories [7]. There have also been works using neural networks, such as Long Short-Term Memory (LSTM) model [8] [9] [10] and Mixture Density Network (MDN) [11]. Many of these give good predictive performance, however, they rely on a large amount of vehicle trajectory data for training, thus are limited to specific scenarios and specific instances of scenarios corresponding to the training data. As a result, these approaches are not generalizable or scalable to many/other scenarios.

There are other works that incorporate dynamics explicitly. For instance, [2] develops a probabilistic anticipation framework for dynamic objects. It adopts a continuous-discrete hybrid state model and describes state probability distribution using a hybrid Gaussian/discrete mixture model (hGMM). And in the continuous stage of mixture propagation, each mixand is designed to split into multiple Gaussian mixands if it fails the linearity criteria, in order to maintain the accuracy of this Gaussian mixture model approximating a non-Gaussian

distribution. [12] proposes a closed-loop two-stage prediction model, where the first stage is a Gaussian Process functional mapping with input as vehicle states and output as control commands and the second is the vehicle dynamics. Since the state distribution is approximated by a Gaussian mixture model, this work investigates Gaussian Process at uncertain inputs, as well as correlation between inputs and outputs, for the dynamics model takes in both vehicle states and control commands. Simulated data of vehicle traversing an intersection are collected for Gaussian Process and the Gaussian splitting method from [2] is adopted for state propagation. This work is limited to data-specific scenarios as well. [13] develops a dynamics-based prediction model and a maneuver-based model, and then combines them with an interaction and mixing approach. Its maneuver estimation is implemented using Dynamic Bayesian Network (DBN) and therefore suffers from the same scalability issues.

1.3 Research Goal and Approach

The goal of this thesis is to develop a framework for anticipation of vehicle motions around an ego vehicle with satisfactory performance and generality. The performance metric can be defined by prediction distance accuracy over certain time period into the future; here this thesis uses a 1.5-second prediction window. And generality is examined by performances of such framework across various scenes, such as straight roadways, intersections and parking garages, as well as across instances of any specific scene, for example, for intersections there are be different configurations, in terms of turning angles, lane widths and number of lanes.

To this end, a general model is proposed, which is referred to as the goal point model. Its main idea is to determine one or several goal points corresponding to the end of prediction period, based on a starting point from a probabilistic tracker and a given road graph. Certain geometric schemes are therefrom designed for cases of vehicle driving straight or making turns. Additional strategies, road partition inference and probabilistic multi-path prediction, are also established for this model to be applied to intersection scenarios. After the goal points are determined, predicted trajectories are propagated with a motion model following interpolated reference paths. This anticipation model is validated via a novel approach using accurate track data from a moving car, and evaluated against an open-loop, purely kinematic baseline model.

1.4 Outline

This thesis is organized as follows. Chapter 2 gives an overview of key elements and definitions for anticipation models, and presents the baseline anticipation model. The goal point anticipation model is demonstrated in Chapter 3, including its core idea and application to both non-intersection roads and intersections. Section 4 describes the data driven validation approach. Experimental results are provided and discussed in Section 5. And Section 6 offers the conclusions.

CHAPTER 2

MODEL ELEMENTS AND DEFINITIONS AND BASELINE ANTICIPATION MODEL

2.1 Overview of Coordinate Systems and Vehicle States

Two coordinate systems are of interest for anticipation models: ego vehicle coordinate system and the inertial coordinate system that is attached to the ground and coincides with the ego vehicle coordinate system at the beginning of prediction. Ego vehicle coordinate is mainly used for instantaneous estimation of vehicle states, and the inertial coordinate is adopted for predicted tracks of vehicles. Figure 2.1 gives a simple illustration of the ego vehicle coordinate system.

In order to make predictions for vehicle movement, knowledge of vehicle current states, or even better, past history of states, is required. In this paper, fundamental vehicle states of interest are: x/y as axial position of target vehicle in ego vehicle coordinate, represented by midpoint of back axle, s as speed of target vehicle in the inertial coordinate, and ϕ as heading of target vehicle in ego vehicle coordinate, all four states shown in Figure 2.1. This thesis uses $\mathbf{q}$ to denote the truth vector of these four states, as $\mathbf{q} = [x\ y\ s\ \phi]^\top$, and $\hat{\mathbf{q}}$ for estimation and $\tilde{\mathbf{q}}$ for prediction. Other vehicle state variables, such as a as acceleration of target vehicle in the inertial coordinate and Ω as angular velocity of target vehicle in the inertial coordinate, may also be utilized in certain anticipation models.

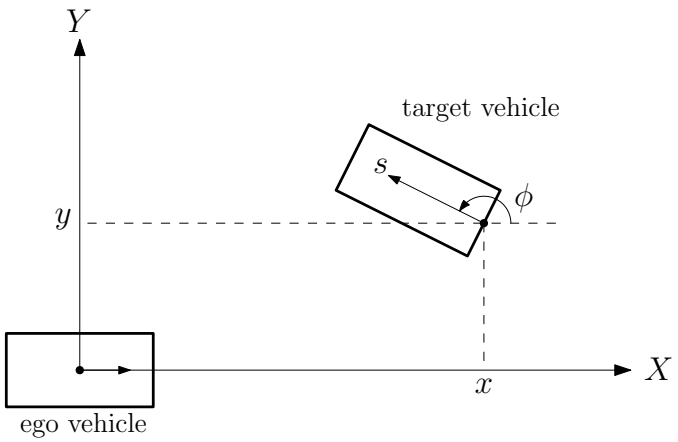


Figure 2.1: Ego Vehicle Coordinate and Target Vehicle States

2.2 Key Environmental Elements for Anticipation

In order for an anticipation model to make accurate prediction of vehicle motions, many environmental elements can go into the model. To list a few, geometric configuration and connections of roads place constraint on vehicle trajectories; stop lines, stop signs and traffic lights greatly affect vehicle motions in terms of acceleration and deceleration; existence and anticipation of other vehicles and pedestrians can also be considered for collision avoidance and so on.

2.3 Baseline Anticipation Model and Its Variations

The baseline anticipation model is a simple open-loop kinematic model. It has several variations, based on different dynamic assumptions about speed and heading. The model can adopt either a constant speed, or a constant acceleration assumption; and it can adopt either a constant heading, or a constant angular velocity assumption. Combination of these assumptions renders four

variations. Other dynamic assumptions can also be chosen as alternatives, such as constant curvature, but they are not addressed in this thesis.

Given the above assumption variations, the corresponding dynamic function can be written as:

$$\dot{\tilde{\mathbf{q}}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{s} \\ \dot{\phi} \end{bmatrix} = \mathbf{f}^p(\tilde{\mathbf{q}}, \mathbf{u}_c) = \begin{bmatrix} s \cdot \cos\phi \\ s \cdot \sin\phi \\ a_c \\ \Omega_c \end{bmatrix}, \tag{2.1}$$

where $\mathbf{u}_c = [a_c \ \Omega_c]^\top$ is the control input vector for assumed acceleration and angular velocity with a_c as either 0 (constant speed) or a given non-zero value (constant acceleration), and Ω_c as either 0 (constant heading) or a given non-zero value (constant angular velocity). A non-zero fixed value for a_c or Ω_c can be determined from estimate of vehicle's current states.

Assume estimate of vehicle's states $\hat{\mathbf{q}}_0$ at time t_0 is acquired from an accurate tracker, for example the one from [14], the baseline model can propagate it using the above dynamic function until time stamp $t_0 + t_p$, where t_p is the prediction time span, to obtain prediction $\tilde{\mathbf{q}}_{t=t_0+t_s:t_0+Nt_s}$, in which N is the corresponding number of time steps and t_s the time interval. This paper only addresses the mean value of predicted states, but uncertainty of prediction can also be added. The continuous-time expression of evolution of corresponding covariance matrix $\tilde{\mathbf{\Sigma}}$ is:

$$\dot{\tilde{\mathbf{\Sigma}}} = \mathbf{F}\tilde{\mathbf{\Sigma}} + \tilde{\mathbf{\Sigma}}\mathbf{F}^\top + \mathbf{B}\mathbf{Q}\mathbf{B}^\top, \tag{2.2}$$

where $\mathbf{F} = \dfrac{\partial \mathbf{f}^p}{\partial \mathbf{q}}$ and $\mathbf{B} = \dfrac{\partial \mathbf{f}^p}{\partial \mathbf{u}_c}$ are Jacobian matrices, and $\mathbf{Q}$ is the covariance matrix for multivariate Gaussian noise linearly added to $\mathbf{u}_c$.

CHAPTER 3

GOAL POINT ANTICIPATION MODEL

3.1 Overview of Goal Point Model

The baseline anticipation model is a very general one. It can be employed at any time in any scenario. It also does not require any additional information. However, its main drawback is that it is road blind, i.e., it does not ensure, or even consider, that the predicted trajectory stays on the road or within the lane, especially when the target vehicle is driving through a curved lane or turning at an intersection. Therefore, the goal point model is proposed here incorporating road information for anticipation of vehicle motions. The general idea of this model is to find a goal point or several goal points with respective probabilities for the end of prediction period based on vehicle's current states, knowledge of road structure and even other environmental features. Figure 3.1 provides a simple case where the ego vehicle is approaching a three-way intersection and makes prediction for two target vehicles. S denotes starting point, namely vehicle's current states, and G goal point.

The following sections demonstrate details on road graph structure used by this model, geometric schemes developed for picking a goal point given a consecutive series of directed road partitions, strategies used to determine goal points based on these geometric schemes for scenarios where vehicles do not traverse intersections and for the one where they do, and interpolation and propagation approaches adopted for track generation.

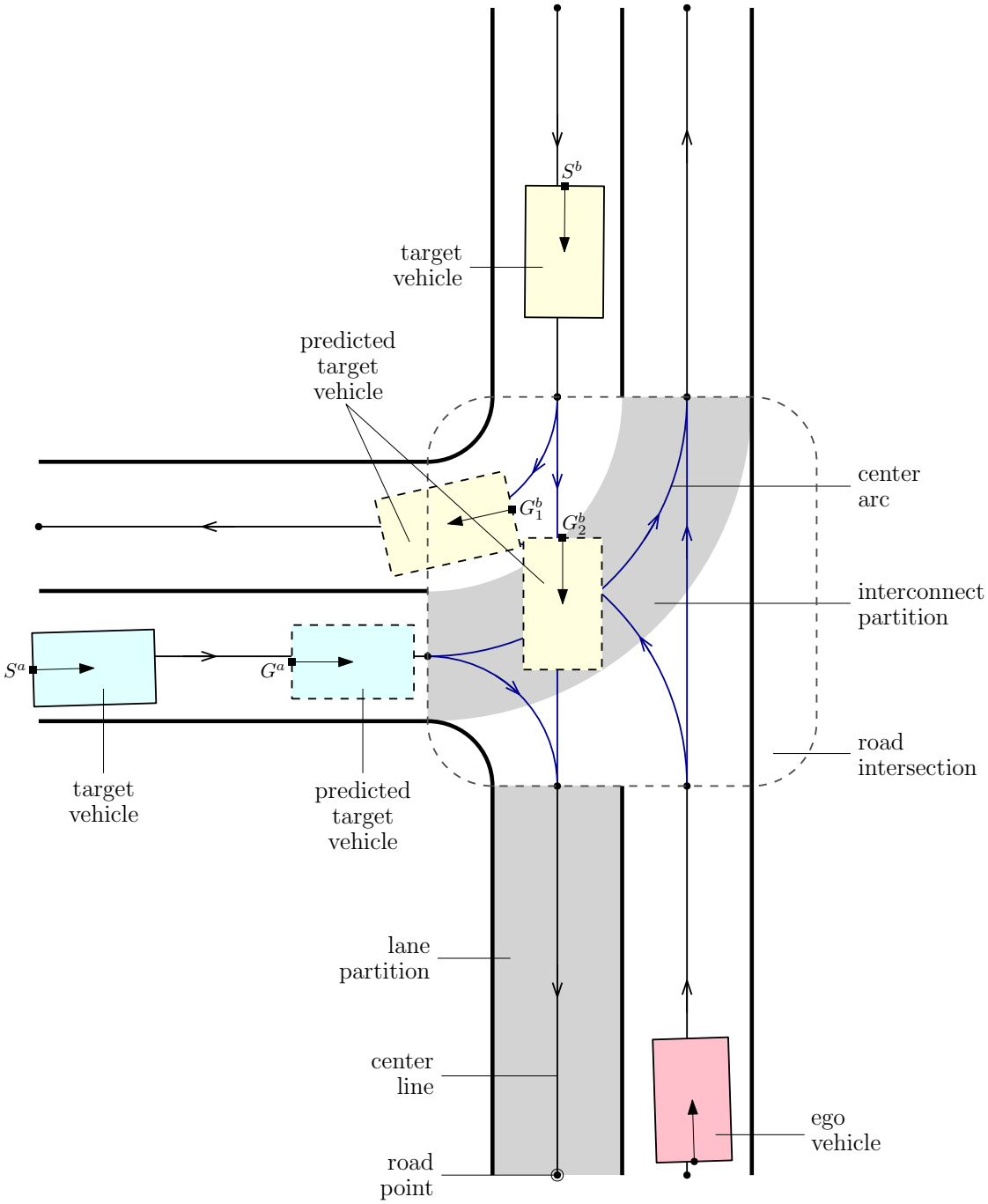


Figure 3.1: Overview of Goal Point Anticipation Model

3.2 Road Graph Structure

This section briefly explains how road information is organized for the goal point model to use. The main element in the road graph structure here is called "road partition", which refers to a physical area on the road, and can be "lane", "interconnect", and "zone". A "lane" partition is a whole representation or a truncation of a normal lane of the road, characterized by the nominal width of the lane, a set of two waypoints denoting the start and the end, and a directed segment in between. An "interconnect" partition connects two lane partitions which are not adjacent to each other (mostly at intersections) and is characterized either by the same way as a lane partition, or by a nominal width, a set of two waypoints denoting the start and the end, and a directed center arc which is tangent to the center segments of both lane partitions this interconnect partition connects and has endpoints with equal distance to the above waypoints on both ends. An interconnect partition is represented by a center arc if it is considered as turning, namely the two lane partitions it connects have a heading difference above a certain threshold (here it's set to 15°). A "zone" partition is an area within which free vehicle movement is permitted, and it's determined by a polygonal boundary using perimeter points. Both waypoint and perimeter point belong to "road point", another element in the road graph structure. Connections among road partitions are illustrated by shared road points. Roads partitions used in this thesis only include lane ones and interconnect ones.

In addition, one more element is further introduced to hold ensembles of road partitions that belong to one single intersection. Named as "road intersection", it specifies every road partition inside the corresponding intersection.

In the case shown in Figure 3.1, examples of road point, lane partition and interconnect partition are highlighted, as well as the intersection, all of whose partition members are depicted with dark blue color.

3.3 Picking a Goal Point Given a Consecutive Series of Road Partitions

Given a series of consecutive road partitions that starts from vehicle's current partition, as well as a desired travel distance, a goal point must be selected for a car traveling down this partition series. This can be further divided into two cases, one where this series does not contain any "turning" road partitions, and the other one where this series does. Schemes for picking goal points for each case are illustrated below.

The desired travel length l used here is a variable for approximating travel distance from the starting point to the goal point. It is calculated with either the assumption of constant speed or constant acceleration, namely, $l = s_0 t_p + \frac{1}{2} a_c t_p^2$, where s_0 is the current speed of vehicle (at the beginning of prediction), and t_p and a_c the same as in Section 2.3. Additionally, the final speed is determined by $s_f = s_0 + a_c t_p$, in accordance with the speed assumption adopted for calculation of the desired travel length.

3.3.1 Case I: Series of Partitions without Turning

In this case, the given series does not contain any turning road partitions. To find the goal point, the vehicle position at the beginning of prediction S is projected onto the center segment of the lane partition it belongs to. The initial offset/distance of starting point to this segment is recorded as d_0. Then the projected point is moved along the connected center segments of corresponding road partitions until the traveled length equals the desired length l. In the end, the projected point is projected back to obtain the goal point G with a decayed offset d_f. The idea of this offset decay comes from the assumption that vehicles are trying to stay in the center of the lane. The decayed final offset is determined using the following equation:

$$d_f = e^{-\alpha l} d_0. \tag{3.1}$$

Here α is a positive scaling factor and can be tuned via collected data. Such approach can be applied to determine parameters of similar equations in this model framework. The final velocity $\vec{v}_f$ is set parallel to the final visited center segment and its magnitude shall be s_f. A demonstration of this scheme is shown in Figure 3.2.

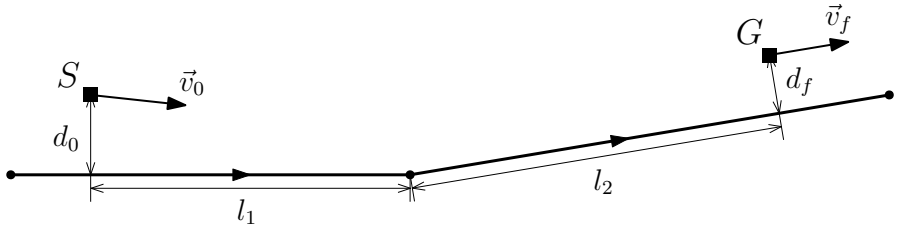


Figure 3.2: Picking Goal Points for Series of Partitions without Turning

3.3.2 Case II: Series of Partitions with Turning

In this case, the given series contains at least one turning partition. To pick the goal point, two schemes are developed and one can be chosen depending on the initial conditions. When the vehicle is not too close toward the intersection, its heading is expected to be within the neighborhood of that of the lane the turning partition follows, thus following the center segments and center arc shall be sufficient. When the vehicle is close enough toward or already inside the intersection, its heading is likely to have significantly differed from that of the lane the turning partition follows and could thus provide more information on how the vehicle is making the turn; so instead of following the center arc new arcs are drawn based on vehicle's current states. These two schemes are named as "center-arc-following" and "dynamic-arc-generating" respectively, and they are discussed as below.

center-arc-following scheme

When the starting point S is not too close (given a distance threshold) toward the interconnect partition of interest, the center-arc-following scheme is adopted. This scheme shares the same idea as that from Subsection 3.3.1, and is demonstrated in Figure 3.3, with the center of the center arc denoted as C. Firstly, the starting point S is projected onto the center segment of current lane partition with an outcoming offset d_0; this partition could be the proceeding partition of the interconnect or even further. Then the projected point is moved along the connected center segments and center arc of these partitions until the traveled length equals the desired length l. Finally the projected point is projected back to obtain the goal point G with a decayed offset d_f, which is same as

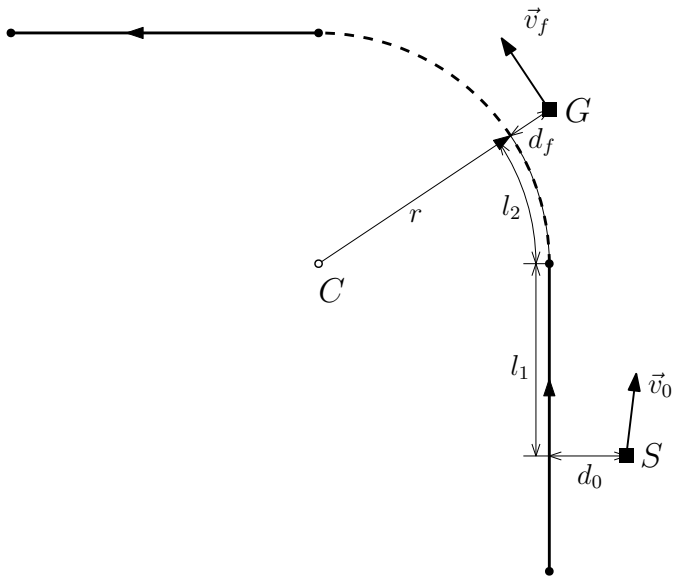


Figure 3.3: Picking Goal Points for Series of Partitions with Turning [Center-Arc-Following]

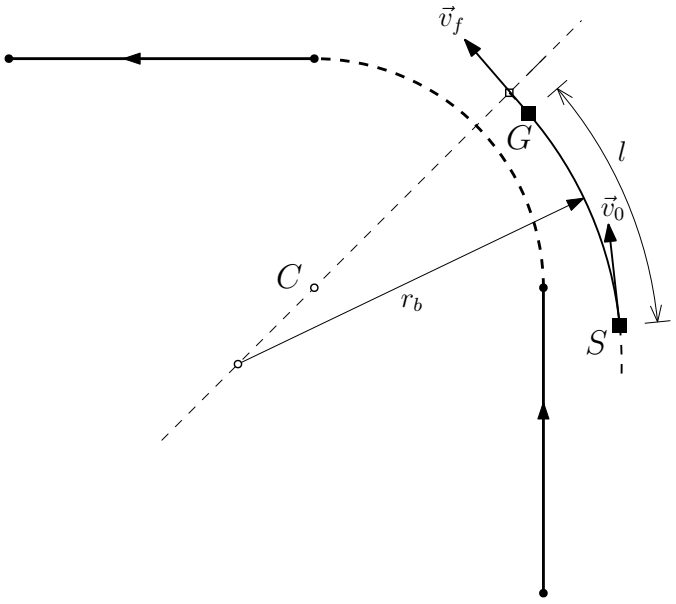
in Equation 3.1. If the final projection is on the center segment of a lane partition, the reverse projection is the same as in Subsection 3.3.1. If the final projection is on the center arc of an interconnect partition, the direction of reverse projection is normal to the arc, and the final velocity $\vec{v}_f$ shall be parallel to the tangent line at the final projection with a magnitude of s_f.

dynamic-arc-generating scheme

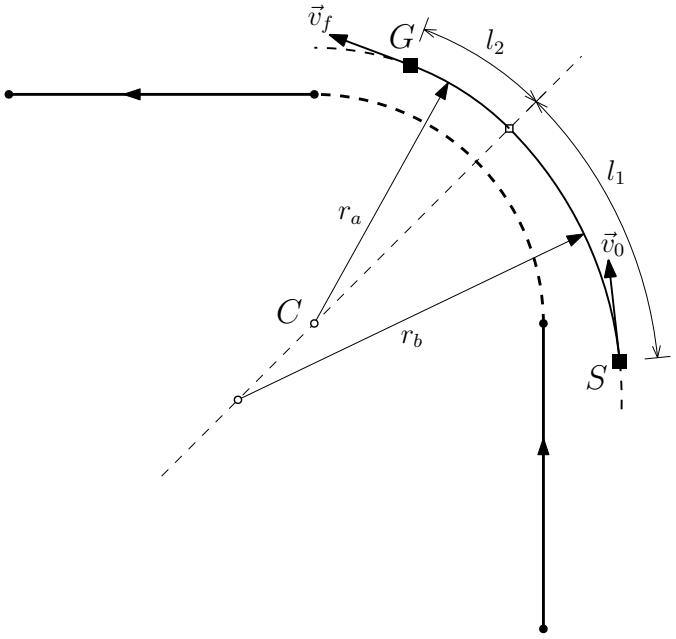
When the starting point S is too close (given a distance threshold) toward or inside the interconnect partition of interest, the scheme is designed to switch to the dynamic-arc-generating one. This approach draws an arc or two arcs based on vehicle's initial states and the bisector about which the interconnect partition's center arc is axial symmetric. Depending on the positions of starting and ending points, there can be several variations, as are illustrated below. Note that if the center of the drawn arc is not on the same side (left or ride) as that of

the center arc relative to the starting point with heading, this scheme is replaced by the center-arc-following scheme.

If S falls before the bisector, as in Figure 3.4, an arc is drawn which centers on the bisector, passes through S, and is tangent to initial velocity $\vec{v}_0$. Then a target point is moved from S along this arc until the traveled length is equal to l. If the target point never crosses the bisector, which is the case of Figure 3.4a, its final position is taken as the goal point G and the final velocity $\vec{v}_f$ shall be tangent to the arc it is on. If the target point ever crosses the bisector, which is the case of Figure 3.4b and Figure 3.4c, right on the bisector it switches to a second arc that centers at C. This arc is designed to serve as a smooth transition from the first arc, due to the fact that following the first arc after the bisector might lead the heading to exceed that of the lane following the turning partition in corresponding direction. Moving on, if the target point does not enter the lane partition following the turning one projection-wise, shown in Figure 3.4b, the principles for choosing goal point G and final velocity $\vec{v}_f$ remain the same. If the target point does enter the following lane partition, shown in Figure 3.4c, the rest of the distance is completed using the approach from Section 3.3.1 to pick the goal point G and determine the final velocity $\vec{v}_f$.

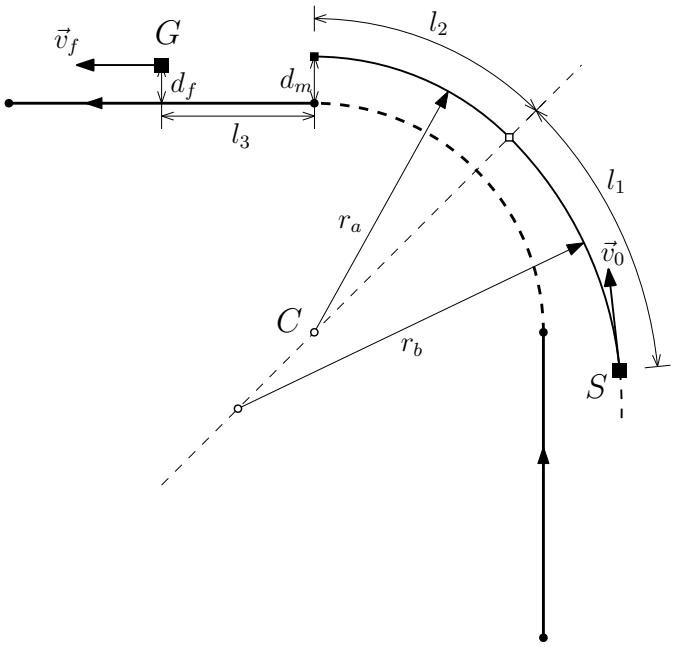


(a)



(b)

Figure 3.4: Picking Goal Points for Series of Partitions with Turning
[Dynamic-Arc-Generating: Starting Before Bisector]

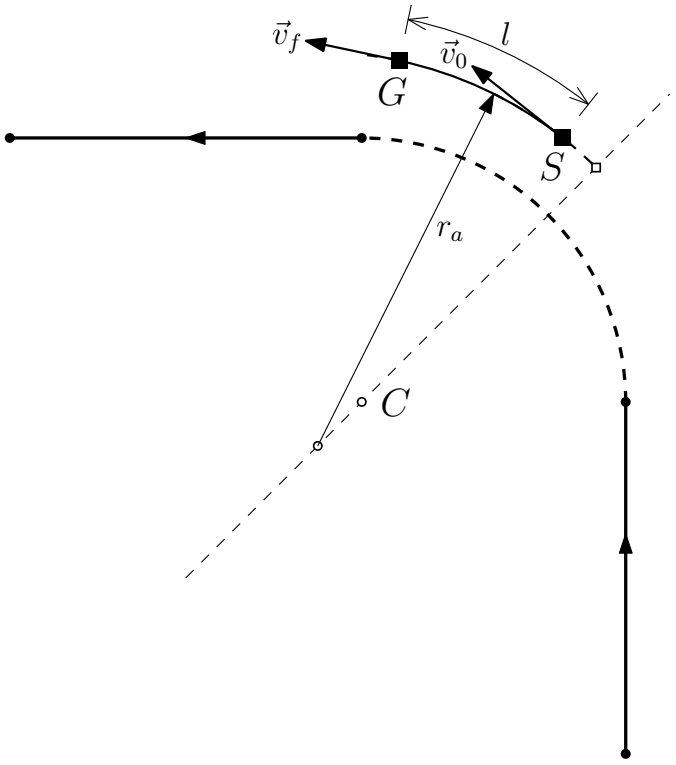


(c)

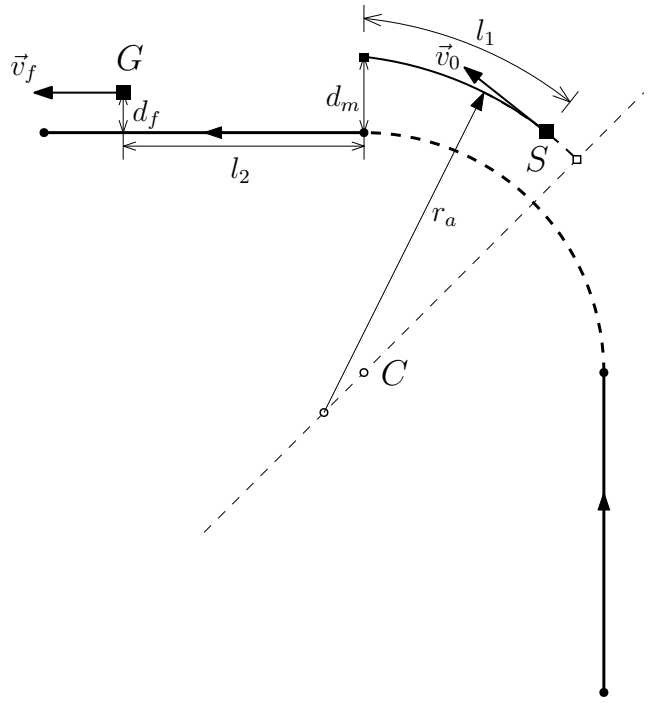
Figure 3.4: Picking Goal Points for Series of Partitions with Turning
[Dynamic-Arc-Generating: Starting Before Bisector]
(Continued)

If S falls after the bisector, as in Figure 3.5, an arc is drawn which centers on the bisector, passes through S, and is tangent to initial velocity $\vec{v}_0$. Then a target point is moved from S along this arc until the traveled length is equal to l. If the target point does not enter the lane partition following the turning one projection-wise, which is the case of Figure 3.5a, its final position is taken as the goal point G and the final velocity $\vec{v}_f$ shall be tangent to the arc it is on. If the target point does enter the following lane partition, which is the case of Figure 3.5b, the rest of the distance is completed using the approach from Section 3.3.1 to pick the goal point G and determine the final velocity $\vec{v}_f$.

Note that for dynamic-arc-generating scheme, the final velocity shall also follow $|\vec{v}_f| = s_f$.



(a)



(b)

Figure 3.5: Picking Goal Points for Series of Partitions with Turning
[Dynamic-Arc-Generating: Starting After Bisector]

3.4 Determination of Goal Points for Non-Intersection Roads

For scenarios where vehicles do not traverse intersections, based on the road graph information, goal points can be easily determined using the schemes demonstrated from the last section. Starting from vehicle's current road partition, a unique consecutive series of directed road partitions can be dynamically established and appended based on partition connections from the road graph, in accordance with the propagation process for picking the goal point. Figure 3.6 presents a special case where the lane in reality is curved and several lane partitions are set to approximate the curve; the above scheme can be easily applied to find the goal point.

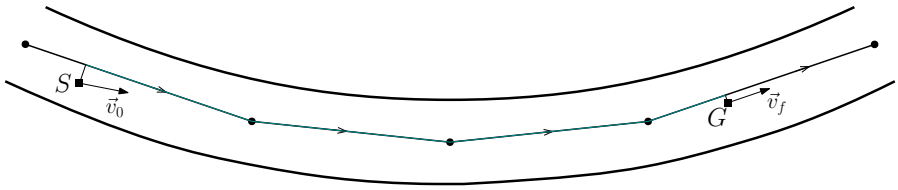


Figure 3.6: Illustration of Determining Goal Point for Curved Lanes

3.5 Determination of Goal Points for Intersection Scenarios

For scenarios where vehicles traverse intersections, there could be more than one possible consecutive series of directed road partitions the target vehicle may visit. To be more specific, two main issues contribute to this concern. One comes from that when a vehicle is spotted at an intersection, which interconnect it belongs to needs to be inferred, and such inference may not be unique. The other rises from different turning decisions the driver is able to make before an

intersection. The first issue entails inference of road partitions at intersections, and both of these issues lead to prediction with multiple goal points.

The following subsections discuss how to infer initial partitions for intersections using a weighted distance metric, and how to determine goal points with probabilities when there are several possible series of road partitions. Moreover, key intersection features can be used to adjust goal points and this model can be modified to be applied to multi-lane cases.

3.5.1 Road Partition Inference at Intersections

When a target vehicle is spotted at an intersection, it is important to figure out which interconnect partition it belongs to, namely, which source lane partition it is driving from and which destination lane partition it is driving into. Such issue can also be seen from the road graph structure developed, in which different interconnect partitions at an intersection can intersect with each other. This problem is tackled using a weighted distance metric to determine possible interconnect partitions.

Given the road partition within which vehicle's current states are, the corresponding intersection can be determined. This algorithm first traverses each interconnect partition inside this intersection. If an interconnect partition is of candidacy, a weighted distance metric is calculated for further inference. An interconnect partition is of candidacy if its nominal heading (center segment heading if it is not a turning partition, or average of starting heading and end heading if it is) is within $\pi/2$ of target vehicle's actual heading. For the i^{th} candidate partition, the target is projected onto the center segment (for non-turning

interconnect partition) or center arc (for turning interconnect partition) as P_i. The distance metric w_i is then calculated as the weighted sum of Euclidean position distance between starting point S and projected point P_i and heading difference between the two, namely:

$$w_i = \alpha_d d_{\langle S,P_i \rangle} + \alpha_h \Delta h_{\langle S,P_i \rangle}, \tag{3.2}$$

where α_d and α_h are corresponding weights, which can also be tuned using collected data.

After weighted distances for all candidate partitions are computed, with the set of indices of these partitions denoted as S_{rp}, the partition with the smallest weighted distance is chosen as the definite possible partition; any partition with a weighted distance no greater than $min(w_i) + w_b$ is also selected, where w_b is a threshold parameter. For the final step, a softmax function is applied to these weighted distances to assign probability p_j for each possible partition, which can be expressed as:

$$p_j = \frac{e^{-w_j}}{\sum\limits_{i \in S_{rp}} e^{-w_i}} \tag{3.3}$$

Figure 3.7 demonstrates this algorithm with an example of three-way intersection. Three interconnect partitions from this intersection are of candidacy. Non-candidate interconnect partitions are colored as gray. The calculated weighted distance of a second partition is within the neighborhood of the smallest weighted distance and this partition is thus chosen, while that of the third partition is beyond the threshold and as a result, this partition is not selected, depicted with a dashed line.

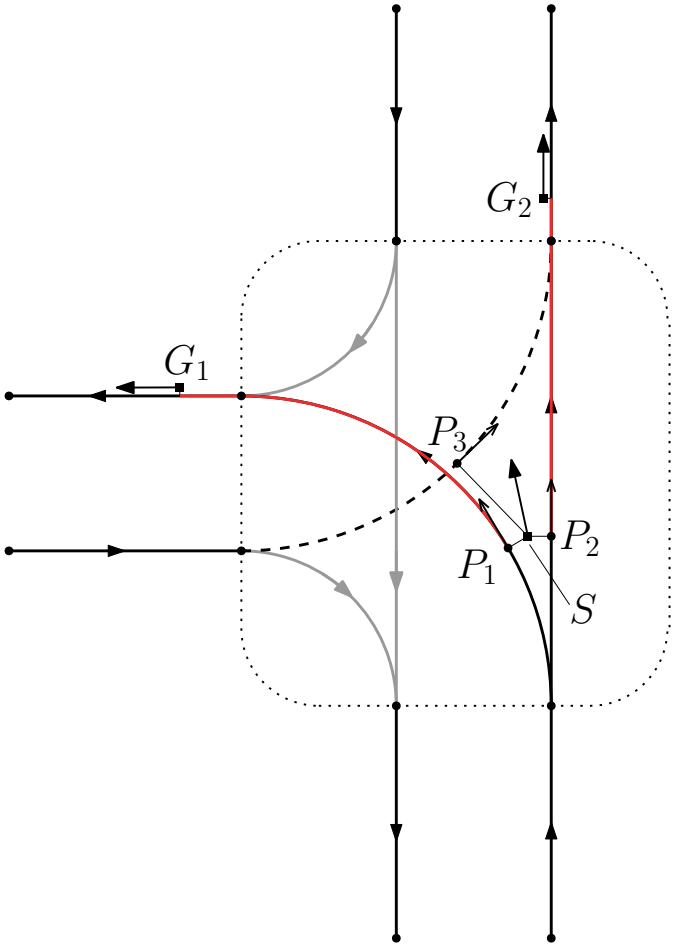


Figure 3.7: Illustration of Scheme for Determining Possible Initial Partitions at Intersections

3.5.2 Multiple Goal Points with Probabilities

Besides the situation discussed in the last subsection, that there are multiple options to choose from when vehicles are approaching intersections can also lead to multiple possible consecutive series of directed road partitions for picking goal points. Thus, the goal point anticipation model is further designed to make predictions incorporating different possible paths.

Starting from a single initial partition, or several initial partitions with probabilities determined from the last subsection, one or more series of road partitions are propagated based on partition connections from the road graph, as well as the process for picking goal points detailed on in Section 3.3. Each series has an initial probability the same as that of its initial partition. When it comes across where its current partition is connected to several next partitions, namely at an intersection, the series splits into several series accordingly, and each new series has an equal probability p_i^j, which sums to that of the original series, p_i. The mathematical expression can be written as:

$$p_i^j = \frac{1}{n} p_i, \tag{3.4}$$

where n is the total number of next partitions, and $j = 1, 2, \cdots, n$. Such propagation stops when the desired travel distance is reached and multiple goal points are thus available.

Figure 3.7 presents an example of multiple goal points G_i for multiple initial partitions. And Figure 3.8 shows a case where a vehicle is driving into a three-way intersection and has two different paths to choose from.

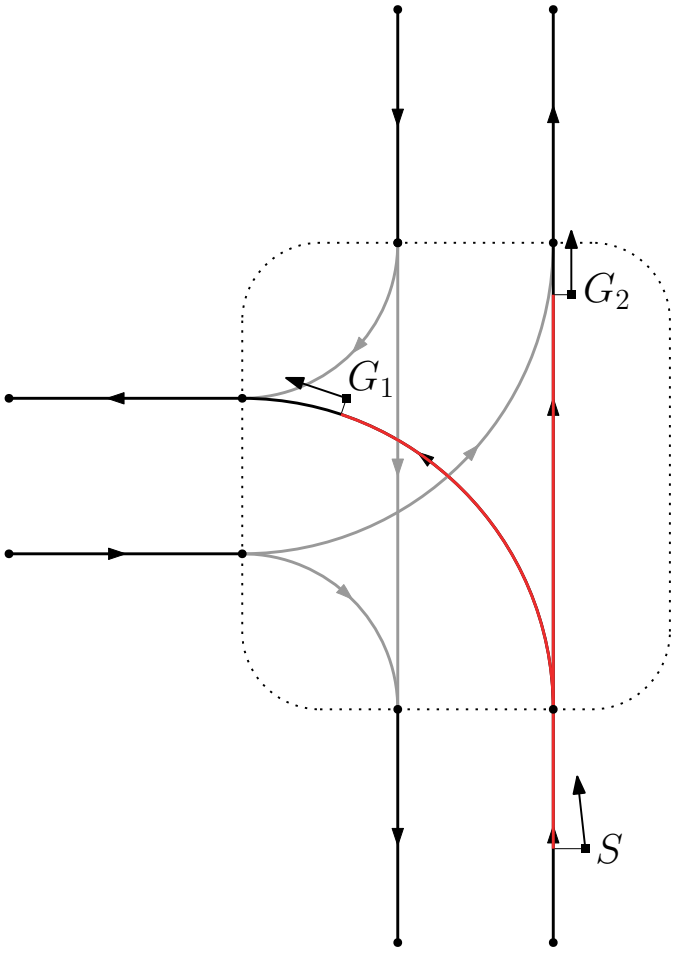


Figure 3.8: Illustration of Multiple Paths for Driving Into an Intersection

3.5.3 Goal Point Adjustment Under Key Intersection Features

There are a few key elements at intersections that can have great influence on driving behavior, especially in terms of acceleration, such as stop lines, stop signs, and traffic lights. Detection of these elements is a known problem and several vision-based techniques exist. To list a few, [15] locates stop lines via a Canny edge detector; [16] presents a shape-based method for detecting road signs, stop sign as an octagon; [17] proposes a two-stage method for traffic light recognition, the first detecting all visible spot lights and the second executing an Adaptive Template Matcher for recognition. Deep-Neural-Network-based approaches have also been investigated, such as [18] and [19]. The existence and motions of other vehicles, cyclists and pedestrians also affect how the target vehicle drives, and can be inferred from tracking of these surrounding objects by the ego vehicle.

With proper detections of stop lines, stop signs, and traffic lights, and tracking of surrounding vehicles, cyclists and pedestrians, it is possible to further adjust goal points to meet corresponding constraints, which can be achieved by assigning a fixed acceleration or deceleration, decreasing desired travel distance so that the vehicle does not enter the intersection when it is supposed not to, or other advanced schemes. Results of preliminary incorporation of these elements into the goal point anticipation model is shown in Section 5.2.

3.5.4 Application to Multi-Lane Roads

The road graph structure used in this goal point model is based on the one Team Cornell used for DARPA Urban Challenge in 2007, in which case all roads

only have one lane for each direction. However, the above strategies for finding goal points at intersections also work for roads with multiple lanes after certain modifications. The road graph shall contain interconnect partitions representing each possible pairing of source lanes and destination lanes. Due to the existence of multiple lanes, it is no longer a good idea to represent a turning interconnect partition with a circular arc. More complicated geometric shapes, such as sheared ellipses, shall be used. The center-arc-following scheme presented in Subsection 3.3.2 needs to be modified to be able to project a point onto a sheared ellipse and move along on it. The dynamic-arc-generating scheme from the same subsection also needs to be modified so that proper sheared ellipse arcs can be generated. Then, the same road partition inference algorithm can be performed to determine which ones of these interconnects are possible initial partitions, and the same probabilistic multi-goal-point scheme can also be applied to different interconnect choices, even if they correspond to the same turning direction.

3.6 Generation of Anticipated Tracks Given Goal Points

Once a goal point or several goal points are determined, certain interpolation and simulation approaches can be employed so that corresponding trajectories with time stamps are available. In this goal point anticipation model, a cubic Hermite spline is used to interpolate from starting point to goal point for each. A closed-loop path following controller is then simulated to enable the car to follow such interpolated paths.

3.6.1 Cubic Hermite Spline

Cubic Hermite spline is a piecewise spline interpolator with each piece as a third-degree polynomial in Hermite form, given values of its two endpoints and their first derivatives. This means it can match the position, speed and heading for start and goal points, i.e., all of the fundamental states. Here the one-piece unit-interval form is adopted, whose expression is:

$$\boldsymbol{p}(t) = (2t^3 - 3t^2 + 1)\boldsymbol{p}_0 + (t^3 - 2t^2 + t)\boldsymbol{m}_0 + (-2t^3 + 3t^2)\boldsymbol{p}_1 + (t^3 - t^2)\boldsymbol{m}_1 \qquad (3.5)$$

where $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$ are values of two endpoints and $\boldsymbol{m}_0$ and $\boldsymbol{m}_1$ first derivatives with $t \in [0, 1]$ (Figure 3.9).

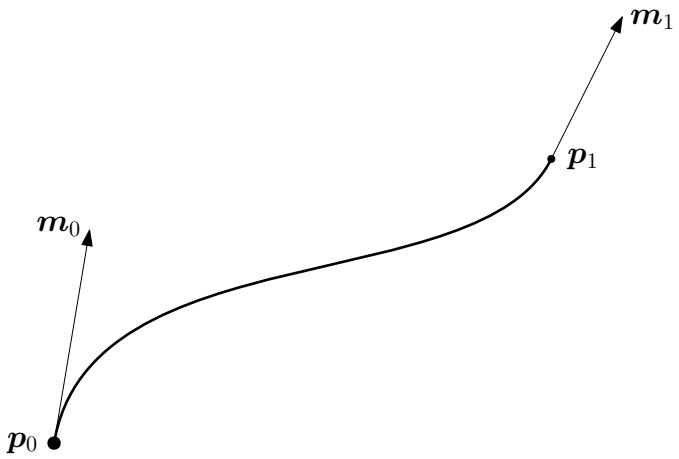


Figure 3.9: Illustration of One-Piece Cubic Hermite Spline

Applying cubic Hermite spline to the goal point anticipation model, $\boldsymbol{p}_0$ is set to the position of the starting point S, $\boldsymbol{p}_1$ the position of the goal point G, $\boldsymbol{m}_0 = \frac{|\overrightarrow{SG}|^2}{\overrightarrow{SG} \cdot \vec{v}_0}\vec{v}_0$, and $\boldsymbol{m}_1 = \frac{|\overrightarrow{SG}|^2}{\overrightarrow{SG} \cdot \vec{v}_f}\vec{v}_f$. This leads to the first derivatives being parallel to starting and final velocities with scaling factors that incorporate length and direction information of starting and ending points and velocities.

3.6.2 Car Simulator

The car simulator used by the goal point model is a path following controller, which is a combination of speed PID controller and steering PID controller given reference path and current and past states, acting on the vehicle dynamics from Equation 2.1. From this car simulator, predicted vehicle tracks can be extracted for each corresponding goal point. Since other objects, like other vehicles and pedestrians, are also being tracked and predicted, simple collision avoidance can be easily added to this path following controller for more realistic anticipation results.

CHAPTER 4

DATA DRIVEN VALIDATION APPROACH

4.1 Overview

This thesis employs a novel approach to generating data for validating performance of anticipation models as well as learning key parameters they use. Given only sensor measurements and ego vehicle odometry, and using an accurate tracker based on the one presented by [14], a set of estimated tracks can be generated with an accuracy of within 1m. This approach uses logged sensor and track data to obtain these tracks and decomposes each track $\hat{\mathbf{q}}_k^a$, denoted as an augmented state vector which may contain more state variables, into several time snippets with length t_p. Each snippet $\tilde{\mathbf{q}}_{k_0+1:k_0+K}$ is transformed from ego vehicle coordinate to inertial coordinate, using posterior pose estimates which is acquired from the algorithm proposed in [20]. Next, this snippet is interpolated based on dynamics as $\tilde{\mathbf{q}}_{t=t_0+t_s:t_0+Nt_s}$ in order to be aligned with a fixed time interval. It is then evaluated against the prediction $\hat{\mathbf{q}}_{t=t_0+t_s:t_0+Nt_s}$ made at the starting time t_0, which is initialized using $\hat{\mathbf{q}}_{k_0}^a$. Euclidean position errors are computed for each time step. For prediction with multiple paths, the one with probability above a given threshold (0.6) is selected; if such path does not exist, the one with the least position error at the end of prediction is chosen. The lower part of Figure 4.1 can be referred to as a summary of how each snippet is processed.

The approach can be scaled to include all vehicle tracks and all times, although our approach uses an interval of 0.5s for start times while picking snippets for each track so as not to have so many "similar" snippets. The time length of prediction is set to 1.5s, and the sampling time for prediction is 0.1s.

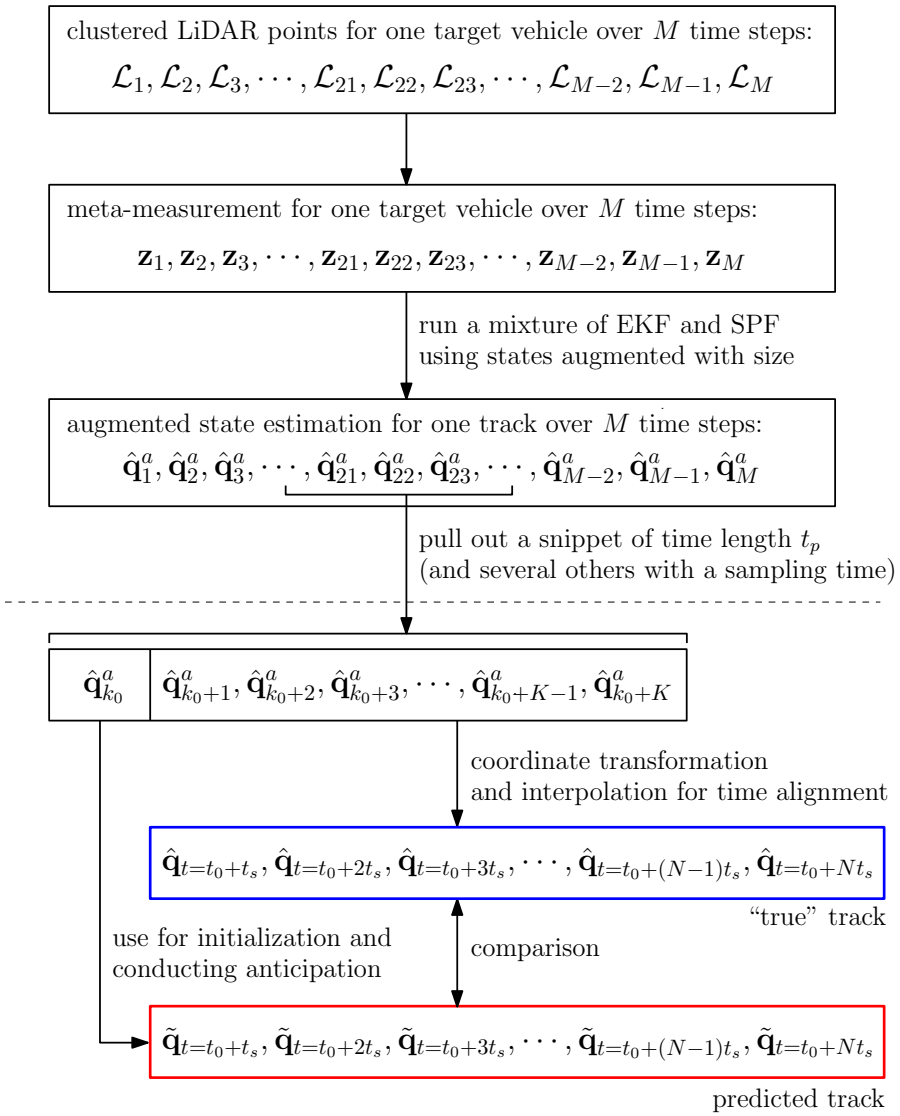


Figure 4.1: Data Flow of Track Estimation and Anticipation Evaluation

The rest of this chapter focuses on details of how each estimated track is generated and how data association is performed given sensor measurements without correspondence.

4.2 Estimation of Vehicle States Using LiDAR

This section talks about how estimation of vehicle states is implemented in this thesis. The upper part of Figure 4.1 shows the data flow for estimation. The general idea is that, assume a time series of clustered 2D LiDAR points for one target vehicle is retrieved, corresponding meta-measurements can be computed and a recursive Bayesian filter is then run with a dynamic model that incorporates vehicle size parameters into states, in order to generate reasonable estimation for vehicle-type objects.

The following subsections walks through the dynamic model, the observation process and what kind of filters are used in detail. Note that all tracking estimates are represented within the instantaneous ego vehicle coordinate system.

Dynamics

Specifically, an augmented state vector is adopted for filtering, as $\mathbf{q}^a = [x\ y\ s\ \phi\ l\ w\ a\ \Omega]^\top$, in which l and w are nominal length and nominal width of the vehicle as it is treated and approximated as a rectangle, a and Ω are acceleration and angular velocity of the vehicle, and the rest are the same as the state variables mentioned in Section 2.1. The vehicle dynamics is modeled with a random Gaussian walk acceleration and a random Gaussian walk angular velocity, for it is assumed that these two states do not change noncontinuously. Derivatives of size parameters are also modeled with Gaussian noises so that they can be estimated towards true nominal values. The state transition func-

tion $\mathbf{f}^a(\mathbf{q}^a, \boldsymbol{\mu}^a)$ for filtering is thus written as:

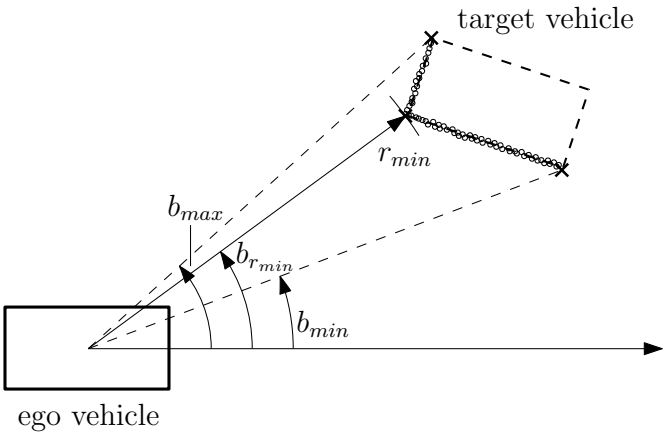
$$\dot{\mathbf{q}}^a = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{s} \\ \dot{\phi} \\ \dot{l} \\ \dot{w} \\ \dot{a} \\ \dot{\Omega} \end{bmatrix} = \mathbf{f}^a(\mathbf{q}^a, \boldsymbol{\mu}^a) = \begin{bmatrix} s \cdot \cos\phi - v_x + \omega_z \cdot y + \mu_x^a \\ s \cdot \sin\phi - v_y - \omega_z \cdot x + \mu_y^a \\ a + \mu_s^a \\ \Omega - \omega_z + \mu_\phi^a \\ \mu_l^a \\ \mu_w^a \\ \mu_a^a \\ \mu_\Omega^a \end{bmatrix} \tag{4.1}$$

where v_x and v_y are components of ego vehicle's velocity in an inertial coordinate attached to the ground along the X-axis and Y-axis of ego vehicle coordinate, ω_z is the rotational speed of ego vehicle in the inertial coordinate, and $\boldsymbol{\mu}^a = [\mu_x^a\ \mu_y^a\ \mu_s^a\ \mu_\phi^a\ \mu_l^a\ \mu_w^a\ \mu_a^a\ \mu_\Omega^a]^\mathsf{T}$ is the process noise vector linearly added to each state.

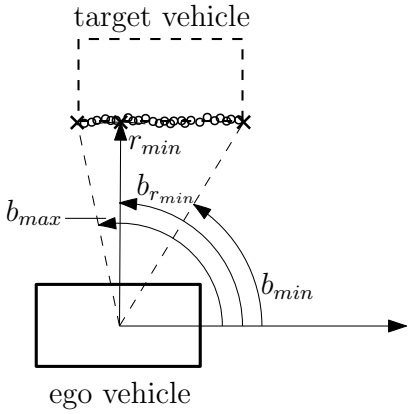
Meta-Measurements and Observation Function

From raw LiDAR point measurements, meta-measurements can be constructed for filtering. The meta-measurement vector contains four elements, as $\mathbf{z} = [b_{min}\ b_{max}\ r_{min}\ b_{r_{min}}]^\mathsf{T}$, in which b_{min} is the minimum bearing counterclockwise, b_{max} the maximum bearing counterclockwise, r_{min} the minimum range, and $b_{r_{min}}$ the corresponding bearing of minimum range. Figure 4.2 shows two cases where the LiDAR sensor detects and collects points from two sides or just one side of the target vehicle.

Given a state vector, the corresponding meta-measurements can be calculated by drawing a rectangle using position, heading and size parameters and determining which side or sides can be detected. Figure 4.3 illustrates two cases



(a) two-side case

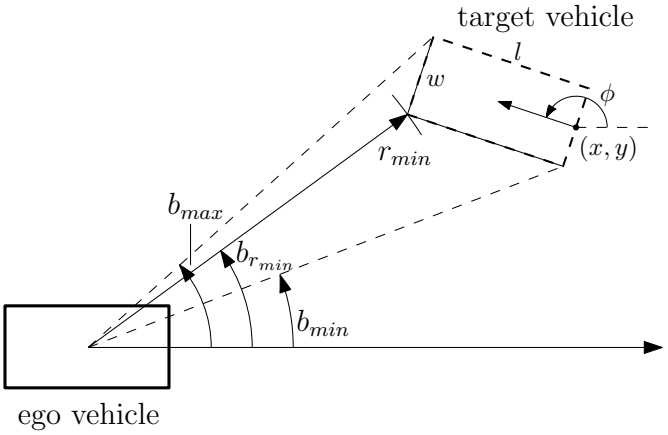


(b) one-side case

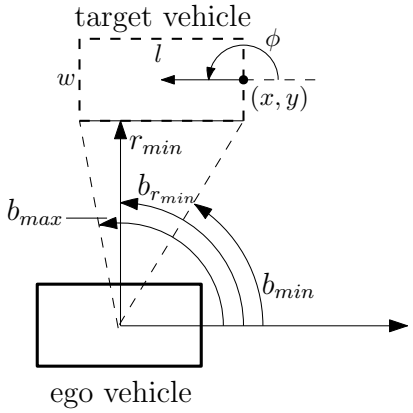
Figure 4.2: Construction of Meta-Measurements from LiDAR Points

where the sensor is expected to be seeing two sides or just one side of the target vehicle. In reality, multiple LiDAR sensors may be actively used for tracking, and each sensor has its own position, orientation, and field of view. This makes the observation process much more complicated. The strategy here is to figure out the complete set of vehicle sides, whole or partial, each sensor can see (this calculation should be done in sensor coordinate), take the union of all sets and place it in ego vehicle coordinate, and compute corresponding meta-measurements. Such observation function for filtering is denoted as $\mathbf{z} = \mathbf{h}(\mathbf{q}^a, \boldsymbol{\nu})$,

where $\boldsymbol{\nu} = [\nu_{b_{min}}\ \nu_{b_{max}}\ \nu_{r_{min}}\ \nu_{b_{r_{min}}}]^{\mathsf{T}}$ is the measurement noise vector linearly added to each meta-measurement.



(a) two-side case



(b) one-side case

Figure 4.3: Calculation of Expected Meta-Measurements

Recursive Bayesian Filtering

Using the dynamic function (Equation 4.1) and the observation function, estimation of vehicle states is done by a mixture of Extended Kalman filter (EKF) and Sigma Point filter (SPF). The prediction step is implemented under EKF, which is not as time consuming as SPF, by calculating the Jacobian matrix of

state transition function, for the dynamic model is simple and not highly non-linear. The update step is implemented under SPF, since the computation of expected meta-measurements needs to distinguish among many cases due to different number of sides the ego vehicle can see from a target vehicle and the multi-sensor situation described above.

4.3 Data Association

In the last section, it is assumed that a time series of clustered 2D LiDAR points for one target vehicle is available. However, in actual operations, raw LiDAR sensor data are fed into a clustering module and separated into clusters with unknown correspondence to each target in real time. Thus data association is required to determine which cluster from one time stamp and which cluster from the next time stamp correspond to the same target.

In this thesis, the algorithm implemented for Cornell's autonomous vehicle Skynet in DARPA Urban Challenge in 2007, which is elaborated in [14], is utilized for data association purpose. Considering the joint estimation problem of both data assignment and obstacle tracking, this algorithm implements a Rao-Blackwellized particle filter (RBPF) to factorize such problem into the two respective one: the data assignment problem tackled with a particle filter, and the obstacle tracking problem with parametric filters. It's worth mentioning that the vehicle states and the dynamic model used for these parametric filters in this original algorithm are different from those adopted in Section 4.2, and do not specifically target at vehicle objects.

Once data assignment is determined from the above algorithm, the filtering

technique illustrated in the last section is conducted, starting from the time series of clustered LiDAR points associated for each target, to generate vehicle estimates.

CHAPTER 5

EXPERIMENTAL EVALUATION

5.1 Details on Software Experiments

In order to evaluate the baseline anticipation model and the goal point anticipation model, software implementation of these models is entailed. Based on the original C++ software framework of Cornell's autonomous vehicle Skynet, new filtering and anticipation modules are written and incorporated into this framework to be run in real time.

The dataset used for evaluation is Team Cornell's data log from DARPA Urban Challenge Event in 2007. Both baseline model and goal point model are run through most of the data log for the first mission, which is approximately 2.5 hours, and nearly 750 snippets are retrieved in total.

A sum of six variations of anticipation models are evaluated, four from the baseline model (constant speed + constant heading, constant acceleration + constant heading, constant speed + constant angular velocity, or constant acceleration + constant angular velocity), and two from the goal point model (constant speed, or constant acceleration).

5.2 Statistic Evaluation Results

In the following comparisons, for each category, the baseline model variation with the best performance is chosen against its counterpart among goal point

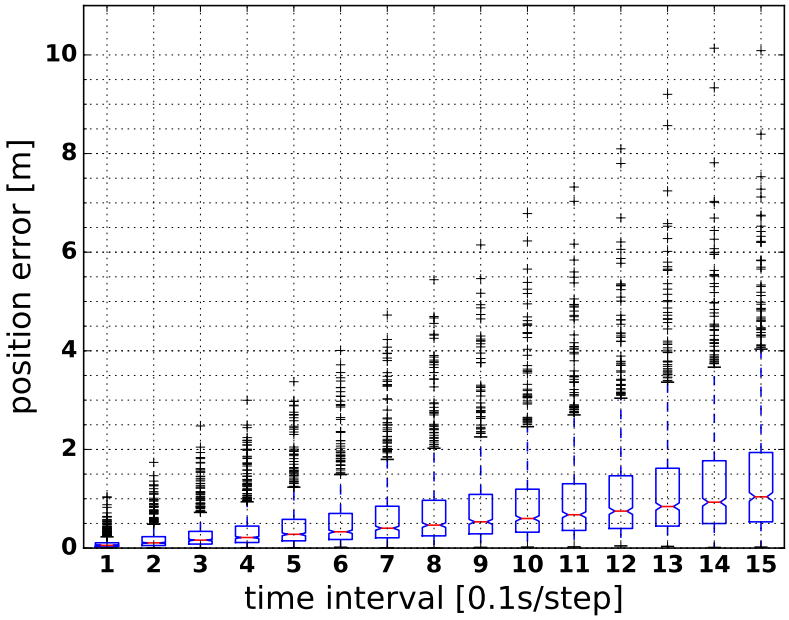
model variations. Box plots are drawn on the evaluated position error results for each model variation at each time step. In these box plots, the central red mark indicates the median, and the bottom and top edges of the blue box represent the 25th and 75th percentiles respectively. The blue dashed lines extend to the black whiskers corresponding to the most extreme data points not considered as outliers, and the outliers are plotted individually using black cross symbols.

Figure 5.1 shows the comparison between the baseline model and the goal point model for straight roadway scenarios. Around 660 snippets are evaluated. It can be seen that the goal point model performs slightly better the the baseline model in terms of median, 25th and 75th percentiles, and upper fence of position errors. Specifically, the goal point model has a median error of 0.80m versus that of baseline model 1.04m, and a 75th percentile 1.66m versus 1.94m, at the final step of prediction.

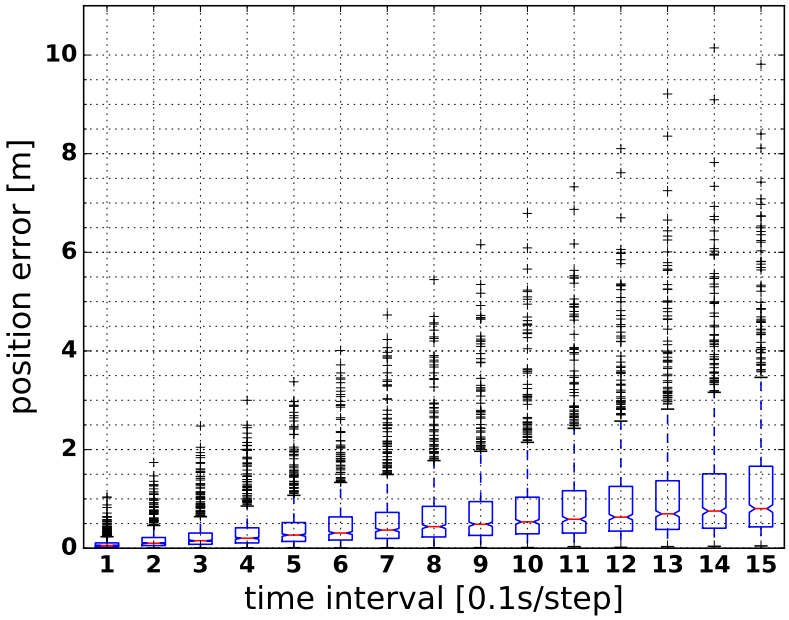
For a more detailed and numeric demonstration of evaluation results, Table 5.1 presents the numeric values of these statistic metrics of interest on position errors of each anticipation model at the final time step. Note that the lower fence is absent due to its negative value.

statistic metrics	anticipation models	
	baseline	goal point
minimum	0.019	0.046
25th percentile	0.053	0.043
median	1.039	0.804
75th percentile	1.940	1.662
upper fence	4.054	3.507
maximum	11.076	11.084

Table 5.1: Statistic Metrics of Position Error at Final Time Step
of Baseline and Goal Point Models for Straight Roadway Scenarios



(a) baseline model



(b) goal point model

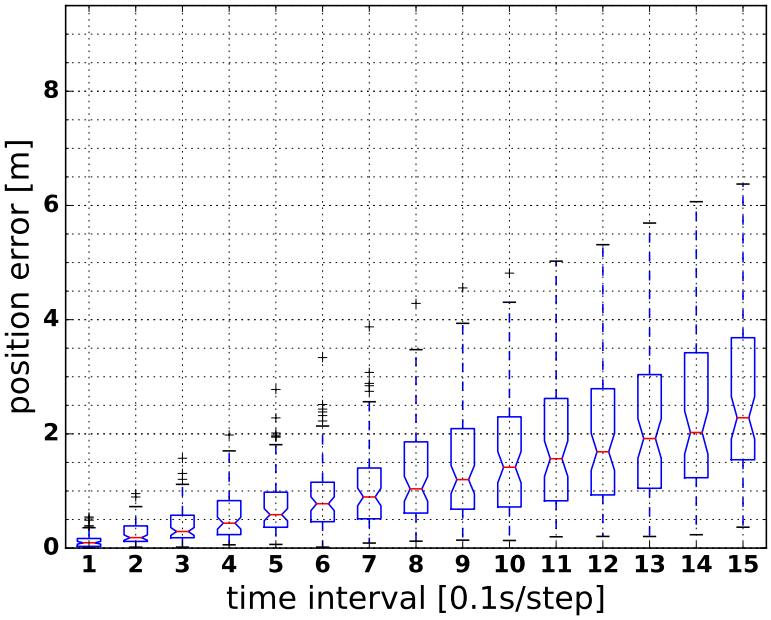
Figure 5.1: Box Plots of Position Errors of Baseline and Goal Point Models for Straight Roadway Scenarios

Figure 5.2 provides the evaluation for intersection and curved lane scenarios, with approximately 90 snippets. Compared with the baseline model, the goal point model has improved the median error from 2.28m to 1.70m by 25.4%, and lowered the 75th percentile error from 3.69m to 3.03m by 17.9%, as well as other statistic metrics like 25th percentile and upper fence, at the final step of prediction.

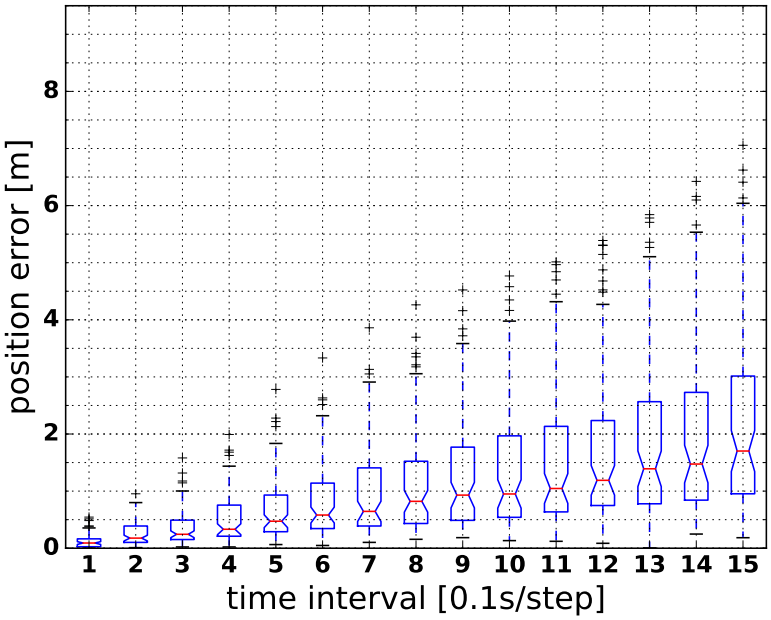
Similar to Table 5.1, Table 5.2 presents the numeric values of these statistic metrics at the final time step.

statistic metrics	anticipation models	
	baseline	goal point
minimum	0.366	0.184
25th percentile	1.511	0.947
median	2.282	1.702
75th percentile	3.686	3.035
upper fence	6.947	6.167
maximum	6.375	7.056

Table 5.2: Statistic Metrics of Position Error at Final Time Step
of Baseline and Goal Point Models for Intersection and Curved Lane Scenarios



(a) baseline model



(b) goal point model

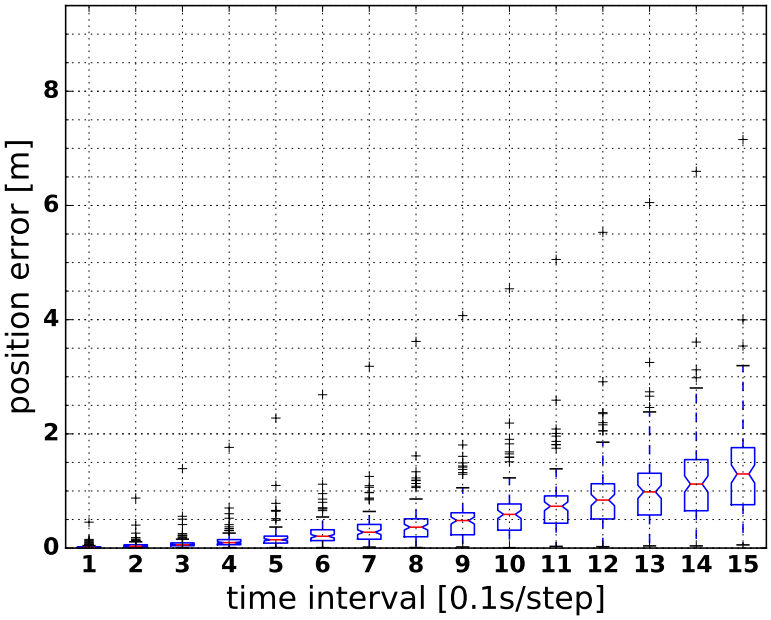
Figure 5.2: Box Plots of Position Errors of Baseline and Goal Point Models for Intersection and Curved Lane Scenarios

In addition, logged data were collected at an intersection in downtown Ithaca, NY, and accurate track data were generated. Key environmental features, such as traffic light changes, other vehicle driving in and out of the way, and pedestrians walking in and out of the way, are labeled for each track as indicators. Then the baseline model and the goal point model are run over this set of data. Figure 5.3 shows the comparison results for whenever such environmental indicators are present. There are around 100 snippts. It can be concluded that the goal point model has significantly better predictive performance over the baseline model under these scenarios. Use the final step for instance, the goal point model has improved the median error by 49.1%, from that of the baseline model 1.40m to 0.71m; the 75th percentile error has also been reduced by 46.3%, from 1.77m to 0.95m.

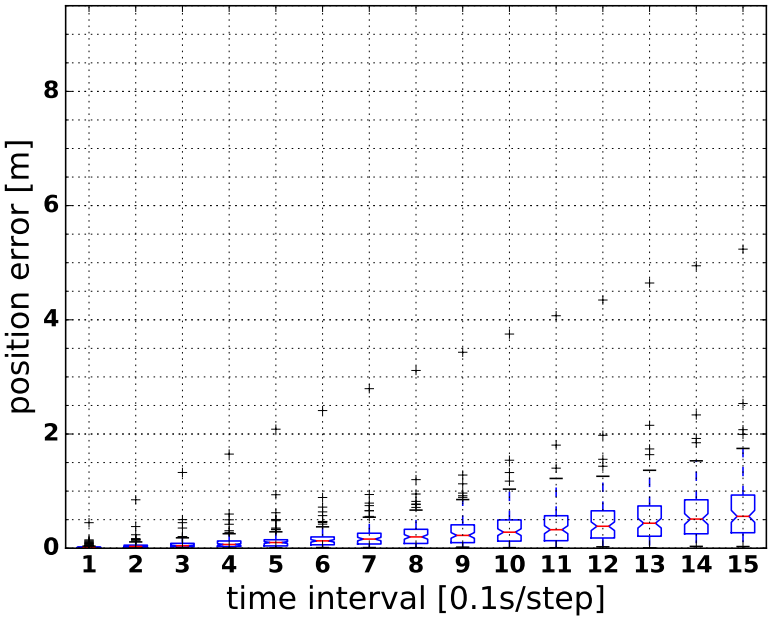
In the same way as Table 5.1, Table 5.3 presents the numeric values of these statistic metrics at the final time step.

statistic metrics	anticipation models	
	baseline	goal point
minimum	0.056	0.029
25th percentile	0.757	0.270
median	1.298	0.558
75th percentile	1.768	0.950
upper fence	3.283	1.969
maximum	7.155	5.237

Table 5.3: Statistic Metrics of Position Error at Final Time Step
of Baseline and Goal Point Models Under Key Environmental Features



(a) baseline model



(b) goal point model

Figure 5.3: Box Plots of Position Errors of Baseline and Goal Point Models Under Key Environmental Features

From the above comparisons, the goal point model has shown improved performance over the baseline model, by 20 – 45% for different scenarios in terms of median position error. Especially for intersection turnings and curved lanes, the goal point model is able to keep the target vehicle on the road or in the lane, which the baseline model cannot guarantee.

Figure 5.4 presents a case where a vehicle is turning left at a three-way intersection. It shows the truth versus predictions at six time stamps. The solid blue line refers to its past true track, dashed blue line its future true track, red line the prediction from the baseline model with constant speed and constant heading, yellow line the prediction from the baseline model with constant speed and constant angular velocity, and teal line the prediction from the goal point model with constant speed. When the vehicle is far from the intersection, it goes straight and all three predictions are similar and match the truth (Figure 5.4a). When the vehicle is approaching the intersection, it has two possible paths to choose from, either going straight or turning left, and the goal point model picks goal point for each, as is shown in Figure 5.4b. When the vehicle just enters the intersection, which is the case of Figure 5.4c, the goal point model infers two possible initial partitions and predicts two corresponding tracks. When the vehicle is further into the intersection, shown in Figure 5.4d, the goal point model is able to determine one unique initial partition, indicating the vehicle is definitely turning right, and its prediction follows this turning behavior perfectly. In the meanwhile, the baseline model with constant speed and constant heading is unable to catch this turning at all; the one with constant speed and constant angular velocity is able to sense this turning but does not perform as well as the goal point model. Figure 5.4e presents when the vehicle is driving out of the intersection; two goal points are determined from the goal point model, one as

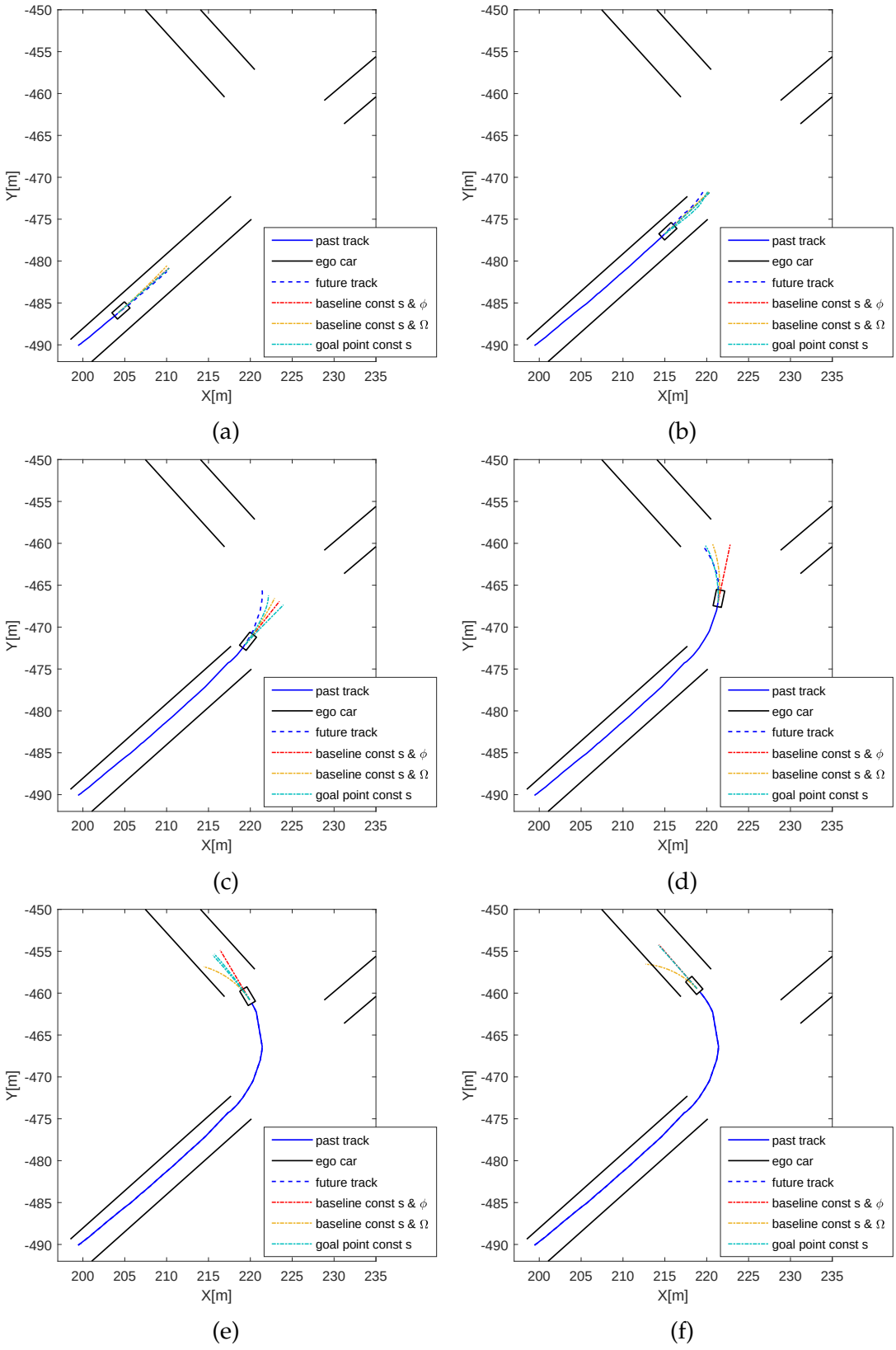


Figure 5.4: Example I of Predictive Performance of Baseline and Goal Point Models

the truth and the other one from the upper right lane turning into the upper left lane, but they share the same destination. In Figure 5.4f, the vehicle has fully adjusted to the final lane partition, and the goal point model and the baseline model with constant speed and constant heading predict its motion well, while the baseline model with constant speed and constant angular velocity still assumes the vehicle is turning and thus drags the prediction off the road.

Figure 5.5 shows a case where a car is approaching an intersection and the traffic light just turns red. The goal point model captures this change and assigns a deceleration in order to move the goal point closer behind the stop line, while the baseline model cannot do so.

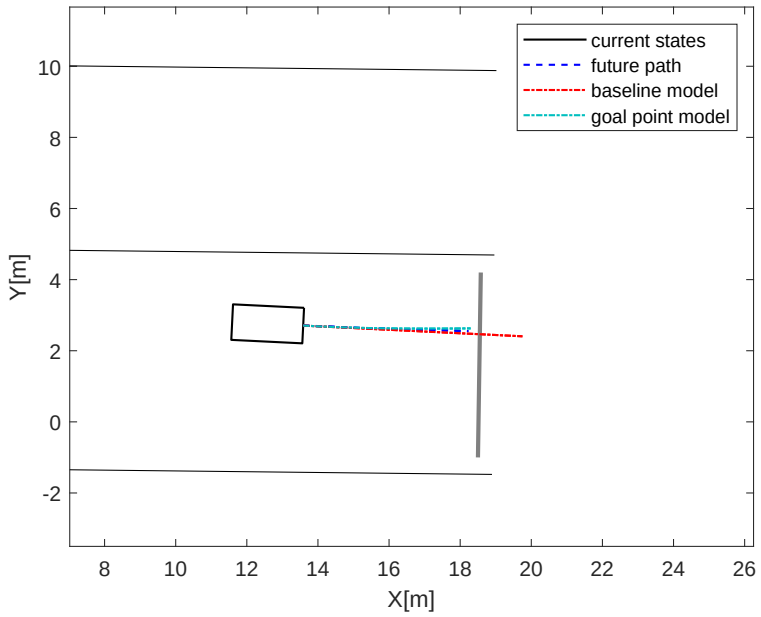


Figure 5.5: Example II of Predictive Performance of Baseline and Goal Point Models

5.3 Analysis of Anticipation Error

From comparing the true estimated states against predicted vehicle tracks for instances with significant position errors, the occasional failure of anticipation can be mainly attributed to two reasons, as are discussed below.

The first one comes from the estimation itself. If any tracked is poorly estimated, not only the anticipation models will receive unsatisfactory initialization, but also the comparison is impaired as the estimated data, which are treated as the truth here, are not close enough to the real truth, making corresponding evaluation unreliable. For example, when a vehicle is turning at an intersection into a destination lane partition, if the filter fails and estimates the vehicle to drive into the adjacent lane, while the goal point model is actually predicting it to drive into the correct lane, evaluation will not consider this as an accurate prediction performance.

The second reason is the speed mismatch. Aside from inaccurate initial speed due to failure of estimation, some vehicles have drastic speed change at intersections. In such case, even models using constant acceleration are unable to capture such characteristic, leading to rather high prediction errors. A comprehensive analysis of vehicle's speed change can be conducted so that more reality-matching assumptions can be made and applied to the goal point anticipation model in order to obtain better prediction results.

CHAPTER 6

CONCLUSION

This thesis proposes a general anticipation framework for making prediction of vehicle motions and trajectories for autonomous driving. This goal point anticipation model aims to find goal points based on road structure and conducts interpolation and simulation to generate predicted tracks. Geometric schemes for picking a goal point given a consecutive series of road partitions with and without turning are designed. Additional strategies are developed for the goal point model to apply the above geometric schemes to intersection scenarios, including initial road partition inference and probabilistic multi-goal-point propagation. Adjusting goal points using key intersection features, such as stop signs and traffic lights, and applying this model to multi-lane intersections are also discussed. This goal point model uses cubic Hermite spline to interpolate from starting points to goal points, and then runs a car simulator, which utilizes PID speed and steering controllers, to generate predicted tracks.

In order to evaluate this anticipation model, the acquisition of data, which are estimates of vehicle states, is enabled by running parametric Bayesian filters on clustered 2D LiDAR points and making use of a data assignment algorithm presented by a previous piece of work. Each vehicle track is decomposed into several snippets, which predictions are made upon by anticipation models and evaluated against. The goal point model along with the baseline model are validated on a real-time data log with LiDAR measurements. According to the experimental evaluation results, the goal point model performs noticeably better than the baseline model under different scenarios.

This thesis has made the following potential contributions:

- It proposes a general model for anticipation of vehicle motions around a moving ego car. This goal point model incorporates map knowledge and can be applied to different scenarios. It also allows room for goal point adjustment using information from other environmental features.
- This goal point model does not rely on any data collected for any specific scenario. However, collected data can be used to tune model parameters and analyze driving behavior in certain cases.
- It adopts a novel approach to generating data for anticipation model validation, which makes use of data from a probabilistic tracker and pose estimates of ego vehicle.

Future goals of this research shall concentrate on 1) obtaining more accurate estimated track data, 2) analyzing acceleration and deceleration behavior of vehicles driving into and away from intersections, and 3) geometric representation for multi-lane intersections and corresponding scheme modifications.

BIBLIOGRAPHY

[1] L. Fletcher, S. Teller, E. Olson, D. Moore, Y. Kuwata, J. How, J. Leonard, I. Miller, M. Campbell, D. Huttenlocher *et al.*, "The mit–cornell collision and why it happened," *Journal of Field Robotics*, vol. 25, no. 10, pp. 775–807, 2008.

[2] F. Havlak and M. Campbell, "Discrete and continuous, probabilistic anticipation for autonomous robots in urban environments," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 461–474, 2014.

[3] S. Sekizawa, S. Inagaki, T. Suzuki, S. Hayakawa, N. Tsuchida, T. Tsuda, and H. Fujinami, "Modeling and recognition of driving behavior based on stochastic switched arx model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 4, pp. 593–606, 2007.

[4] Y. Kishimoto and K. Oguri, "A modeling method for predicting driving behavior concerning with driver's past movements," in *Vehicular Electronics and Safety, 2008. ICVES 2008. IEEE International Conference on*. IEEE, 2008, pp. 132–136.

[5] R. Hamada, T. Kubo, K. Ikeda, Z. Zhang, T. Bando, and M. Egawa, "A comparative study of time series modeling for driving behavior towards prediction," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*. IEEE, 2013, pp. 1–4.

[6] B. Tang, S. Khokhar, and R. Gupta, "Turn prediction at generalized intersections," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015, pp. 1399–1404.

[7] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, "Probabilistic trajectory prediction with gaussian mixture models," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, 2012, pp. 141–146.

[8] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 399–404.

[9] F. Altché and A. de La Fortelle, "An lstm network for highway trajectory prediction," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 353–359.

[10] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1179–1184.

[11] Y. Hu, W. Zhan, and M. Tomizuka, "Probabilistic prediction of vehicle semantic intention and motion," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 307–313.

[12] J. Hardy, F. Havlak, and M. Campbell, "Multi-step prediction of nonlinear gaussian process dynamics models with adaptive gaussian mixtures," *The International Journal of Robotics Research*, vol. 34, no. 9, pp. 1211–1227, 2015.

[13] G. Xie, H. Gao, L. Qian, B. Huang, K. Li, and J. Wang, "Vehicle trajectory prediction by integrating physics-and maneuver-based approaches using interactive multiple models," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5999–6008, 2018.

[14] I. Miller, M. Campbell, and D. Huttenlocher, "Efficient unbiased tracking of multiple dynamic obstacles under large viewpoint changes," *IEEE Transactions on Robotics*, vol. 27, no. 1, pp. 29–46, 2011.

[15] I. Miller, M. Campbell, D. Huttenlocher, F.-R. Kline, A. Nathan, S. Lupashin, J. Catlin, B. Schimpf, P. Moran, N. Zych *et al.*, "Team cornell's skynet: Robust perception and planning in an urban environment," *Journal of Field Robotics*, vol. 25, no. 8, pp. 493–527, 2008.

[16] G. Loy and N. Barnes, "Fast shape-based road sign detection for a driver assistance system," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 1. IEEE, 2004, pp. 70–75.

[17] R. De Charette and F. Nashashibi, "Traffic light recognition using image processing compared to learning processes," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 333–338.

[18] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, "No need to worry about adversarial examples in object detection in autonomous vehicles," *arXiv preprint arXiv:1707.03501*, 2017.

[19] K. Behrendt, L. Novak, and R. Botros, "A deep learning approach to traffic lights: Detection, tracking, and classification," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1370–1377.

[20] I. Miller, M. Campbell, and D. Huttenlocher, "Map-aided localization in sparse global positioning system environments using vision and particle filtering," *Journal of Field Robotics*, vol. 28, no. 5, pp. 619–643, 2011.